In-Class Lab 9

ECON 425 (Justin Heflin, West Virginia University)

March 24, 2023

The purpose of this lab is to practice using R to test for serial correlation and how to fix it. The lab may be completed as a group. To receive credit, upload your .R script to the appropriate place on eCampus ("In-Class Labs'' folder).

For starters

Open a new R script (named `ICL9_XYZ.R`, where `XYZ` are your initials)

Clean out/"Sweep'' R Studio

Click the broom in the Environment panel (top-right), it is directly below the Tutorial button. Also, in the bottom-right panel, click the Plots button and then click the broom in that panel. This should help with loading things into R.

R Packages

First, install the `pdfetch`, `tsibble`, and `COVID19` packages. `pdfetch` stands for ''Public Data Fetch" and is a slick way of downloading statistics on stock prices, GDP, inflation, unemployment, etc. `tsibble` is a package useful for working with time series data. `COVID19` pulls up-to date data on COVID-19 cases, deaths, etc.

```
install.packages("COVID19")
install.packages("sandwich")
install.packages("lmtest")
install.packages("pdfetch")
install.packages("tsibble")
install.packages("magrittr")
install.packages("tidyverse)
```

```
library(wooldridge)
library(car)
library(magrittr)
library(lmtest)
library(pdfetch)
library(sandwich)
library(tsibble)
library(COVID19)
library(tidyverse)
```

Load the data

We're going to use data on `US COVID-19` cases, death, and other information

```
COVID_Data <- covid19(c("US"))
COVID_Time_Series <- as_tsibble(COVID_Data, key = id, index = date)
```

Now it will be easy to include lags of various varibles into our regression models

```
View(COVID_Time_Series)
```

Plot time series data

Let's have a look at the data on **new** daily cases and deaths:

```
COVID_Time_Series %<>% mutate(new_deaths = difference(deaths), # data comes in cumulative format
                  new_tests  = difference(tests), # "difference" converts it to
                  new_cases  = difference(confirmed)) # "new cases" etc.

# plots
ggplot(COVID_Time_Series, aes(date, new_cases)) + geom_line()
```

```
## Warning: Removed 19 rows containing missing values (`geom_line()`).
```

```
ggplot(COVID_Time_Series, aes(date, new_deaths)) + geom_line()
```

```
## Warning: Removed 19 rows containing missing values (`geom_line()`).
```

```
# plots with 7-day rolling average
ggplot(COVID_Time_Series, aes(date, new_cases)) + geom_line(aes(y=rollmean(new_cases, 7,
                                                                 na.pad=TRUE)))
```

```
## Warning: Removed 25 rows containing missing values (`geom_line()`).
```

```
ggplot(COVID_Time_Series, aes(date, new_deaths)) + geom_line(aes(y=rollmean(new_deaths, 7,
                                                                  na.pad=TRUE)))
```

```
## Warning: Removed 25 rows containing missing values (`geom_line()`).
```

Determinants of US COVID-19 Cases

Now let's estimate the following regression model:

$$\log(new\_cases_t) = \beta_0 + \beta_1 gath_t + \beta_2 gath_{t-7} + \beta_3 gath_{t-14} + \beta_4 \log(new\_cases_{t-7}) + u_t$$

where new_cases is the number of new COVID cases, and $gath$ is a variable taking on values 0–4 representing severity of gatherings restricitons.

```
COVID_Time_Series %<>% mutate(log.new.cases = log(new_cases),
                  log.new.cases = replace(log.new.cases,new_cases==0,NA_real_))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `log.new.cases = log(new_cases)`.
## Caused by warning in `log()`:
## ! NaNs produced
```

```
regression <- lm(log.new.cases ~ gatherings_restrictions + lag(gatherings_restrictions,7) +
                          lag(gatherings_restrictions,14) + lag(log.new.cases,7),
                 data=COVID_Time_Series)
summary(regression)
```

```
##
## Call:
## lm(formula = log.new.cases ~ gatherings_restrictions + lag(gatherings_restrictions,
##     7) + lag(gatherings_restrictions, 14) + lag(log.new.cases,
##     7), data = COVID_Time_Series)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.87003 -0.18663 -0.02246  0.18695  1.96682
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      1.400710   0.082420  16.995  < 2e-16 ***
## gatherings_restrictions         -0.262853   0.035866  -7.329 4.68e-13 ***
## lag(gatherings_restrictions, 7)   0.012143   0.053460   0.227     0.82
## lag(gatherings_restrictions, 14)  0.255587   0.035775   7.144 1.71e-12 ***
## lag(log.new.cases, 7)            0.876298   0.007888 111.097  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4115 on 1033 degrees of freedom
##   (138 observations deleted due to missingness)
## Multiple R-squared:  0.9286, Adjusted R-squared:  0.9283
## F-statistic:  3357 on 4 and 1033 DF,  p-value: < 2.2e-16
```

Testing for Serial Correlation

Using the Durbin-Watson Test:

```
durbinWatsonTest(regression)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.4509892      1.086493       0
##  Alternative hypothesis: rho != 0
```

Now using the Lagrange Multiplier Test:

```r
bgtest(regression)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  regression
## LM test = 211.85, df = 1, p-value < 2.2e-16
```

Correcting for Serial Correlation

Now let's compute Newey-West standard errors. To do so, we'll use the `vcov` option in the `modelsummary()` function along with the `NeweyWest` function from the `sandwich` package.

```r
NW_VCOV <- NeweyWest(regression)
coeftest(regression, vcov = NW_VCOV)
```

```
##
## t test of coefficients:
##
##                                    Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                        1.400710   0.323312  4.3324 1.619e-05 ***
## gatherings_restrictions           -0.262853   0.104689 -2.5108   0.01220 *
## lag(gatherings_restrictions, 7)    0.012143   0.085269  0.1424   0.88679
## lag(gatherings_restrictions, 14)   0.255587   0.106085  2.4093   0.01616 *
## lag(log.new.cases, 7)              0.876298   0.028811 30.4150 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How does your interpretation of the the effect of gathering restrictions change after using the Newey-West standard errors?